

Data Analysis for E-Commerce Challenge

Data yang digunakan

Dataset yang digunakan merupakan data dari DQLab Store yang merupakan e-commerce dimana pembeli dan penjual saling bertemu. Pengguna bisa membeli barang dari pengguna lain yang berjualan. Setiap pengguna bisa menjadi pembeli sekaligus penjual.

Ada 4 tabel yang tersedia,

1. users, berisi detail data pengguna. Berisi,
 - user_id : ID pengguna
 - nama_user : nama pengguna
 - kodepos : kodepos alamat utama dari pengguna
 - email : email dari pengguna
2. products, berisi detail data dari produk yang dijual. Berisi,
 - product_id : ID produk
 - desc_product : nama produk
 - category : kategori produk
 - base_price : harga asli dari produk
3. orders, berisi transaksi pembelian dari pembeli ke penjual. Berisi,
 - order_id : ID transaksi
 - seller_id : ID dari pengguna yang menjual
 - buyer_id : ID dari pengguna yang membeli
 - kodepos : kodepos alamat pengiriman transaksi (bisa beda dengan alamat utama)
 - subtotal : total harga barang sebelum diskon
 - discount : diskon dari transaksi
 - total : total harga barang setelah dikurangi diskon, yang dibayarkan pembeli
 - created_at : tanggal transaksi
 - paid_at : tanggal dibayar
 - delivery_at : tanggal pengiriman
4. order_details, berisi detail barang yang dibeli saat transaksi. Berisi,
 - order_detail_id : ID table ini
 - order_id : ID dari transaksi
 - product_id : ID dari masing-masing produk transaksi
 - price : harga barang masing-masing produk
 - quantity : jumlah barang yang dibeli dari masing-masing produk

10 Transaksi terbesar user 12476

Membuat SQL Query agar menampilkan 10 transaksi dari pembelian dari pengguna dengan user_id 12476, urutkan dari nilai transaksi paling besar. Tampilkan variabel seller_id, buyer_id, nilai_transaksi, dan tanggal_transaksi

SQL Query :

```
1 select seller_id, buyer_id, total as nilai_transaksi, created_at as  
   tanggal_transaksi  
2 from orders  
3 where buyer_id = 12476  
4 order by 3 desc  
5 limit 10 ;
```

Maka tampilan seperti ini :

seller_id	buyer_id	nilai_transaksi	tanggal_transaksi
61	12476	12014000	2019-12-23 00:00:00
53	12476	9436000	2019-12-05 00:00:00
64	12476	4951000	2019-12-19 00:00:00
57	12476	4854000	2019-12-01 00:00:00
22	12476	4010000	2019-11-29 00:00:00
48	12476	1440000	2020-02-27 00:00:00
61	12476	1053000	2019-10-17 00:00:00
35	12476	816000	2020-05-12 00:00:00
60	12476	740000	2019-09-26 00:00:00
3	12476	399000	2019-09-26 00:00:00

Transaksi per bulan

Membuat SQL Query agar menampilkan summary transaksi per bulan di tahun 2020.

Tampilkan variabel tahun_bulan, jumlah_transaksi, dan total_nilai_transaksi.

SQL Query :

```
1 select EXTRACT(YEAR_MONTH FROM created_at) as tahun_bulan, count(1) as  
   jumlah_transaksi, sum(total) as total_nilai_transaksi  
2 from orders  
3 where created_at>='2020-01-01'  
4 group by 1  
5 order by 1 ;
```

Maka tampilan seperti ini :

tahun_bulan	jumlah_transaksi	total_nilai_transaksi
202001	5062	9941756800
202002	5872	12665113550
202003	7323	17189378400
202004	7955	21219233750
202005	10026	31288823000

Pengguna dengan rata-rata transaksi terbesar di Januari 2020

Membuat SQL Query agar menampilkan 10 pembeli dengan rata-rata nilai transaksi terbesar yang bertransaksi minimal 2 kali di Januari 2020. Tampilkan variabel buyer_id, jumlah_transaksi, dan avg_nilai_transaksi.

SQL Query :

```
1 select buyer_id, count(1) as jumlah_transaksi, avg(total) as
   avg_nilai_Transaksi
2 from orders
3 where created_at>='2020-01-01' and created_at<'2020-02-01'
4 group by 1
5 having count(1)>= 2
6 order by 3 desc
7 limit 10
```

Maka tampilan seperti ini :

buyer_id	jumlah_transaksi	avg_nilai_Transaksi
11140	2	11719500.0000
7905	2	10440000.0000
12935	2	8556500.0000
12916	2	7747000.0000
17282	2	6797500.0000
1418	2	6741000.0000
5418	2	5336000.0000
11906	2	5309500.0000
12533	2	5218500.0000
841	2	5052500.0000

Transaksi besar di Desember 2019

Membuat SQL Query agar menampilkan semua nilai transaksi **minimal** 20,000,000 di bulan Desember 2019, tampilkan nama_pembeli, nilai transaksi dan tanggal_transaksi, urutkan sesuai abjad pembeli.

SQL Query :

```
1 select nama_user as nama_pembeli, total as nilai_transaksi, created_at as
   tanggal_transaksi
2 from orders
3 inner join users on buyer_id = user_id
4 where created_at>='2019-12-01' and created_at<'2020-01-01'
5 and total >= 20000000
6 order by 1|
```

Maka tampilan seperti ini :

nama_pembeli	nilai_transaksi	tanggal_transaksi
Diah Mahendra	21142000	2019-12-24 00:00:00
Dian Winarsih	22966000	2019-12-21 00:00:00
dr. Yulia Waskita	29930000	2019-12-28 00:00:00
drg. Kajen Siregar	27893500	2019-12-10 00:00:00
Drs. Ayu Lailasari	22300000	2019-12-09 00:00:00
Hendri Habibi	21815000	2019-12-19 00:00:00
Kartika Habibi	25760000	2019-12-22 00:00:00
Lasmanto Natsir	22845000	2019-12-27 00:00:00
R.A. Betania Suryono	20523000	2019-12-07 00:00:00
Syahrini Tarihoran	29631000	2019-12-05 00:00:00
Tgk. Hamima Sihombing, M.Kom.	29351400	2019-12-25 00:00:00
Tgk. Lidya Lazuardi, S.Pt	20447000	2019-12-16 00:00:00

Kategori Produk Terlaris di 2020

Menampilkan 5 Kategori dengan total quantity terbanyak di tahun 2020, hanya untuk transaksi yang sudah terkirim ke pembeli. Tampilkan category, total_quantity, total_price.

SQL Query :

```
1 SELECT
2     category,
3     SUM(quantity) AS total_quantity,
4     SUM(price) AS total_price
5 FROM
6     orders
7     INNER JOIN order_details USING(order_id)
8     INNER JOIN products USING(product_id)
9 WHERE
10     created_at >= '2020-01-01'
11     AND delivery_at IS NOT NULL
12 GROUP BY
13     1
14 ORDER BY
15     2 DESC
16 LIMIT
17     5;
```

Maka tampilan seperti ini :

category	total_quantity	total_price
Kebersihan Diri	944018	1333153000
Fresh Food	298372	793756000
Makanan Instan	280481	67868000
Bahan Makanan	218151	120563000
Minuman Ringan	212103	63017000

Mencari pembeli high value

Membuat SQL query untuk mencari pembeli yang sudah bertransaksi lebih dari 5 kali, dan setiap transaksi lebih dari 2,000,000.

Tampilkan nama_pembeli, jumlah_transaksi, total_nilai_transaksi, min_nilai_transaksi.
Urutkan dari total_nilai_transaksi terbesar!

SQL Query :

```
1 SELECT
2     nama_user AS nama_pembeli,
3     COUNT(1) AS jumlah_transaksi,
4     SUM(total) AS total_nilai_transaksi,
5     MIN(total) AS min_nilai_transaksi
6 FROM
7     orders
8     INNER JOIN users ON buyer_id = user_id
9 GROUP BY
10     user_id,
11     nama_user
12 HAVING
13     COUNT(1) > 5
14     AND MIN(total) > 2000000
15 ORDER BY
16     3 DESC;
```

Maka tampilan seperti ini :

nama_pembeli	jumlah_transaksi	total_nilai_transaksi	min_nilai_transaksi
Dr. Sidiq Thamrin	6	41976000	2088000
Dina Lailasari	8	38195000	2736000
Puput Uyainah	6	32710000	2677000
R. Tirta Nasyidah	6	25117800	2308800
Martani Laksmiwati	6	24858000	2144000
Fitria Narpati	6	22820000	2337000

Mencari Dropshipper

Mencari tahu pengguna yang menjadi dropshipper, yakni pembeli yang membeli barang akan tetapi dikirim ke orang lain. Ciri-cirinya yakni transaksinya banyak, dengan alamat yang berbeda-beda.

Membuat SQL query untuk mencari pembeli dengan 10 kali transaksi atau lebih yang alamat pengiriman transaksi selalu berbeda setiap transaksi.

Tampilkan nama_pembeli, jumlah_transaksi, distinct_kodepos, total_nilai_transaksi, avg_nilai_transaksi. Urutkan dari jumlah transaksi terbesar.

SQL Query :

```
1 SELECT
2 nama_user AS nama_pembeli,
3 COUNT(1) AS jumlah_transaksi,
4 COUNT(DISTINCT orders.kodepos) AS distinct_kodepos,
5 SUM(total) AS total_nilai_transaksi,
6 AVG(total) AS avg_nilai_transaksi
7 FROM
8 orders
9 INNER JOIN users ON buyer_id = user_id
10 GROUP BY
11 user_id,
12 nama_user
13 HAVING
14 COUNT(1) >= 10
15 AND COUNT(1) = COUNT(DISTINCT orders.kodepos)
16 ORDER BY
17 2 DESC;
```

Maka tampilan seperti ini :

nama_pembeli	jumlah_transaksi	distinct_kodepos	total_nilai_transaksi	avg_nilai_transaksi
Anastasia Gunarto	10	10	7899000	789900.0000
R.M. Setya Waskita	10	10	30595000	3059500.0000

Mencari Reseller Offline

Selanjutnya, akan dicari tahu jenis pengguna yang menjadi reseller offline atau punya toko offline, yakni pembeli yang sering sekali membeli barang dan seringnya dikirimkan ke alamat yang sama. Pembelian juga dengan quantity produk yang banyak. Sehingga kemungkinan barang ini akan dijual lagi.

Membuat SQL query untuk mencari pembeli yang punya 8 tau lebih transaksi yang alamat pengiriman transaksi sama dengan alamat pengiriman utama, dan rata-rata total quantity per transaksi lebih dari 10.

Tampilkan nama_pembeli, jumlah_transaksi, total_nilai_transaksi, avg_nilai_transaksi, avg_quantity_per_transaksi. Urutkan dari total_nilai_transaksi yang paling besar.

SQL Query :

```
1 SELECT
2     nama_user AS nama_pembeli,
3     COUNT(1) AS jumlah_transaksi,
4     SUM(total) AS total_nilai_transaksi,
5     AVG(total) AS avg_nilai_transaksi,
6     AVG(total_quantity) AS avg_quantity_per_transaksi
7 FROM
8     orders
9     INNER join users ON buyer_id = user_id
10    INNER JOIN (
11        SELECT
12            order_id,
13            SUM(quantity) AS total_quantity
14        FROM
15            order_details
16        GROUP BY
17            1
18    ) AS summary_order USING(order_id)
19 WHERE
20     orders.kodepos = users.kodepos
21 GROUP BY
22     user_id,
23     nama_user
24 HAVING
25     COUNT(1) >= 8
26     AND AVG(total_quantity) > 10
27 ORDER BY
28     3 DESC;
```


Maka tampilan seperti ini :

nama_pembeli	jumlah_transaksi	total_nilai_transaksi	avg_nilai_transaksi	avg_quantity_per_transaksi
Gandi Rahmawati	12	36822000	3068500.0000	65.1667
Tgk. Cengkal Hutasoit, M.Ak	8	33943000	4242875.0000	91.7500
Prima Usamah	8	24576000	3072000.0000	67.2500
Ir. Paris Siregar, S.Sos	8	22512000	2814000.0000	87.7500
Raisa Habibi, S.Ked	8	17344000	2168000.0000	54.7500
R. Prima Laksmiwati, S.Gz	8	17269000	2158625.0000	50.1250
Lulut Pangestu, S.Ked	8	16580000	2072500.0000	79.2500
Dt. Radika Winarno	8	16320000	2040000.0000	45.2500
Kayla Astuti	12	15953250	1329437.5000	40.3333
Ajiman Haryanti	8	15527000	1940875.0000	47.5000
Laila Jailani	10	14412000	1441200.0000	56.2000
Jaka Hastuti	8	13914000	1739250.0000	39.5000
Anastasia Safitri	8	13188000	1648500.0000	51.7500
Dr. Ganep Latupono, S.Kom	8	12443200	1555400.0000	70.0000
Luhung Pradipta	8	11272000	1409000.0000	65.6250
R.M. Gaiman Setiawan	8	11268000	1408500.0000	71.5000
Ir. Shania Padmasari, S.Psi	10	10839200	1083920.0000	44.0000
Wirda Aryani	8	10580000	1322500.0000	36.7500
Drs. Indah Megantara, M.TI.	8	10176000	1272000.0000	47.2500
T. Empluk Mandasari, S.IP	8	10066500	1258312.5000	48.5000
Among Nugroho	8	9996000	1249500.0000	30.2500
R. Tania Pangestu	8	9414000	1176750.0000	47.7500
Cakrawangsa Prasetyo, S.T.	10	7966000	796600.0000	28.6000
dr. Galuh Hastuti, S.IP	8	4352000	544000.0000	26.7500
Oni Halim	8	2472000	309000.0000	26.0000

Pembeli sekaligus penjual

Membuat SQL query untuk mencari penjual yang juga pernah bertransaksi sebagai pembeli minimal 7 kali.

Tampilkan nama_pengguna, jumlah_transaksi_beli, jumlah_transaksi_jual. Urutkan berdasarkan abjad dari nama_pengguna.

SQL Query :

```
1 SELECT
2     nama_user AS nama_pengguna,
3     jumlah_transaksi_beli,
4     jumlah_transaksi_jual
5 FROM
6     users
7     INNER JOIN (
8         SELECT
9             buyer_id,
10            COUNT(1) AS jumlah_transaksi_beli
11        FROM
12            orders
13        GROUP BY
14            1
15    ) AS buyer ON buyer_id = user_id
16    INNER JOIN (
17        SELECT
18            seller_id,
19            COUNT(1) AS jumlah_transaksi_jual
20        FROM
21            orders
22        GROUP BY
23            1
24    ) AS seller ON seller_id = user_id
25 WHERE
26     jumlah_transaksi_beli >= 7
27 ORDER BY
28     1;
```

Maka tampilan seperti ini :

nama_pengguna	jumlah_transaksi_beli	jumlah_transaksi_jual
Bahuwiryah Haryanto	8	1032
Bahuwiryah Haryanto	8	1032
Dr. Adika Kusmawati, S.Pt	7	1098
Dr. Adika Kusmawati, S.Pt	7	1098
Gandi Rahmawati	8	1078
Gandi Rahmawati	8	1078
Jaka Hastuti	7	1094
Jaka Hastuti	7	1094
R.M. Prayogo Damanik, S.Pt	8	1044
R.M. Prayogo Damanik, S.Pt	8	1044

Pembeli sekaligus penjual

Membuat SQL query untuk mencari penjual yang juga pernah bertransaksi sebagai pembeli minimal 7 kali.

Tampilkan nama_pengguna, jumlah_transaksi_beli, jumlah_transaksi_jual. Urutkan berdasarkan abjad dari nama_pengguna.

SQL Query :

```
1 SELECT
2     nama_user AS nama_pengguna,
3     jumlah_transaksi_beli,
4     jumlah_transaksi_jual
5 FROM
6     users
7     INNER JOIN (
8         SELECT
9             buyer_id,
10            COUNT(1) AS jumlah_transaksi_beli
11        FROM
12            orders
13        GROUP BY
14            1
15    ) AS buyer ON buyer_id = user_id
16    INNER JOIN (
17        SELECT
18            seller_id,
19            COUNT(1) AS jumlah_transaksi_jual
20        FROM
21            orders
22        GROUP BY
23            1
24    ) AS seller ON seller_id = user_id
25 WHERE
26     jumlah_transaksi_beli >= 7
27 ORDER BY
28     1;
```

Maka tampilan seperti ini :

nama_pengguna	jumlah_transaksi_beli	jumlah_transaksi_jual
Bahuwirya Haryanto	8	1032
Bahuwirya Haryanto	8	1032
Dr. Adika Kusmawati, S.Pt	7	1098
Dr. Adika Kusmawati, S.Pt	7	1098
Gandi Rahmawati	8	1078
Gandi Rahmawati	8	1078
Jaka Hastuti	7	1094
Jaka Hastuti	7	1094
R.M. Prayogo Damanik, S.Pt	8	1044
R.M. Prayogo Damanik, S.Pt	8	1044

Lama transaksi dibayar

Ingin diketahui bagaimana trend lama waktu transaksi dibayar sejak dibuat.

Membuat SQL Query untuk menghitung rata-rata lama waktu dari transaksi dibuat sampai dibayar, dikelompokkan per bulan.

Tampilkan tahun_bulan, jumlah_transaksi, avg_lama_dibayar, min_lama_dibayar, max_lama_dibayar. Urutkan berdasarkan tahun_bulan.

SQL Query :

```
1 SELECT
2     EXTRACT(
3         YEAR_MONTH
4         FROM
5             created_at
6     ) AS tahun_bulan,
7     COUNT(1) AS jumlah_transaksi,
8     AVG(DATEDIFF(paid_at, created_at)) AS avg_lama_dibayar,
9     MIN(DATEDIFF(paid_at, created_at)) min_lama_dibayar,
10    MAX(DATEDIFF(paid_at, created_at)) max_lama_dibayar
11 FROM
12     orders
13 WHERE
14     paid_at IS NOT NULL
15 GROUP BY
16     1
17 ORDER BY
18     1;
```

Maka tampilan seperti ini :

tahun_bulan	jumlah_transaksi	avg_lama_dibayar	min_lama_dibayar	max_lama_dibayar
201901	117	7.0467	1	14
201902	354	7.5399	1	14
201903	668	7.4602	1	14
201904	984	7.2910	1	14
201905	1462	7.3692	1	14
201906	1913	7.5729	1	14
201907	2667	7.4549	1	14
201908	3274	7.6216	1	14
201909	4327	7.5021	1	14
201910	5577	7.4706	1	14
201911	7162	7.5188	1	14
201912	10131	7.4980	1	14
202001	5062	7.4152	1	14
202002	5872	7.5092	1	14
202003	7323	7.4674	1	14
202004	7955	7.4792	1	14
202005	10026	7.4549	1	14

Data Mentor

